Applicant: MacLellean et al
U.S.S.N.: 10/654,852
Filing Date: September 4, 2003
EMC Docket No.: EMC-03-066

In the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the Application.

Listing of Claims:

1. (Presently amended) A method for operating a memory system comprising:

A. receiving a digital word having N bits of data and M bits for error detection;

B. generating a first error correction code based on the N bits of data of the digital word;

C. generating a second error correction code based on the N bits of data of the digital word;

D. performing a first logic operation on the first error correction code and the second error correction code to generate a data signature representative of a comparison of the first error correction code and the second error correction code;

E. performing a second logic operation on the data signature and the M bits of the digital word to generate a constant signal representing a comparison of the data signature and the M bits of the digital word;

F. comparing the generated constant signal to a predetermined constant signal to determine if an error has occurred in at least one of the N bits of data in the digital word, the M bits of data in the digital word, the first error correction code and the second error correction code; and

G. determining that an error has occurred in ~~at least one of~~ the N bits of data in the digital word, the M bits of data in the digital word, the first error correction code ~~and~~ or the second error correction code if the generated constant signal is different from the predetermined constant signal.

2. (Previously presented) The method of claim 1 wherein the first error correction code includes Y bits, wherein each one of the Y bits is generated by performing a third logic operation on predetermined bit sets of the digital word.

3. (Previously presented) The method of claim 2 wherein at least two of the Y bits of the first error correction code are generated by performing the third logic operation on different predetermined bit sets of the digital word.

4. (Previously presented) The method of claim 3 wherein the second error correction code includes Y bits, wherein each one of the Y bits of the second error correction code is generated by performing a fourth logic operation on bits of the digital word which are not included in the predetermined bit set of the digital word used to generate a corresponding bit of the first error correction code.

5. (Previously presented) The method of claim 1 wherein the M bits of the digital word for error correction are parity bits.

6. (Presently amended) The method of claim 4 wherein the predetermined constant signal comprises Z bits, all of which being one of either all zeros ~~and~~ or all ones.

7. (Presently amended) The method of claim 6 wherein the generated constant signal comprises Z bits and, if no errors are present in ~~at least one of~~ the N bits of data in the digital word, the M bits of data in the digital word, the first error correction code ~~and~~ or the second error correction code, all of the Z bits of the generated constant signal are ~~one of~~ either all zeros ~~and~~ or all ones.

8. (Previously presented) The method of claim 7 wherein the first logic operation is an XOR operation.

9. (Previously presented) The method of claim 8 wherein the second logic operation is an XOR operation.

10. (Previously presented) The method of claim 9 wherein the third logic operation is one of an XOR operation and an XNOR operation.

11. (Previously presented) The method of claim 10 wherein the fourth logic operation is one of an XOR operation and an XNOR operation.

12. (Presently amended) The method of claim 4 wherein if no errors are present in the N bits of the digital word, the second error correcting code is ~~one of~~ either identical to the first error correcting code ~~and opposite~~ or complement to the first error correcting code.

13. (Presently amended) The method of claim 4 wherein if no errors are present in the N bits of the digital word, the digital signature comprises ~~one of~~ either all zeros ~~and~~ or all ones.

14. (Previously presented) The method of claim 13 wherein if the N bits of the digital word are even, and if no errors are present in the N bits of the digital word, the digital signature comprises all zeros.

15. (Previously presented) The method of claim 13 wherein if the N bits of the digital word are odd, and if no errors are present in the N bits of the digital word, the digital signature comprises all ones.

16. (Previously presented) The method of claim 12 wherein if the N bits of the digital word are even, and if no errors are present in the N bits of the digital word, the second error correcting code is identical to the first error correcting code.

17. (Presently amended) The method of claim 12 wherein if the N bits of the digital word are odd, and if no errors are present in the N bits of the digital word, the second error correcting code is ~~opposite~~ the complement of the first error correcting code.

18. (Presently amended) A memory system comprising:

an input portion for receiving a digital word having N bits of data and M bits for error detection;

a first error correction code generator for generating a first error correction code based on the N bits of data of the digital word;

a second error correction code generator for generating a second error correction code based on the N bits of data of the digital word;

a first logic operator for performing a first logic operation on the first error correction code and the second error correction code to generate a data signature representative of a comparison of the first error correction code and the second error correction code;

a second logic operator for performing a second logic operation on the data signature and the M bits of the digital word to generate a constant signal representing a comparison of the data signature and the M bits of the digital word; and

a comparator for comparing the generated constant signal to a predetermined constant signal and generating an error signal indicating that an error has occurred in ~~at least one of~~ the N bits of data in the digital word, the M bits of data in the digital word, the first error correction code ~~and~~ or the second error correction code.

19. (Previously presented) The system of claim 18 wherein the first error correction code includes Y bits, wherein the first error correction code generator generates each one of the Y by performing a third logic operation on predetermined bit sets of the digital word.

20. (Previously presented) The system of claim 19 wherein the first error correction code generator generates at least two of the Y bits of the first error correction code by performing the third logic operation on different predetermined bit sets of the digital word.

21. (Previously presented) The system of claim 20 wherein the second error correction code includes Y bits, wherein the second error correction code generator generates each one of the Y bits of the second error correction code by performing a fourth logic operation on bits of the digital word which are not included in the predetermined bit set of the digital word used to generate a corresponding bit of the first error correction code.

22. (Previously presented) The system of claim 18 wherein the M bits of the digital word for error correction are parity bits.

23. (Presently amended) The system of claim 21 wherein the predetermined constant signal comprises Z bits, all of which being ~~one of~~ either all zeros ~~and~~ or all ones.

24. (Presently amended) The system of claim 23 wherein the generated constant signal comprises Z bits and, if no errors are present in ~~at least one of~~ the N bits of data in the digital word, the M bits of data in the digital word, the first error correction code ~~and~~ or the second error correction code, all of the Z bits of the generated constant signal are ~~one of~~ either all zeros ~~and~~ or all ones.

25. (Previously presented) The system of claim 24 wherein the first logic operation is an XOR operation.

26. (Previously presented) The system of claim 25 wherein the second logic operation is an XOR operation.

27. (Previously presented) The system of claim 26 wherein the third logic operation is one of an XOR operation and an XNOR operation.

28. (Previously presented) The system of claim 27 wherein the fourth logic operation is one of an XOR operation and an XNOR operation.

29. (Presently amended) The method of claim 21 wherein if no errors are present in the N bits of the digital word, the second error correcting code is ~~one of~~ either identical to the first error correcting code ~~and opposite~~ or complement to the first error correcting code.

30. (Presently amended)The system of claim 21 wherein if no errors are present in the N bits of the digital word, the digital signature comprises ~~one of~~ either all zeros ~~and~~ or all ones.

31. (Previously presented) The system of claim 30 wherein if the N bits of the digital word are even, and if no errors are present in the N bits of the digital word, the digital signature comprises all zeros.

32. (Previously presented) The system of claim 30 wherein if the N bits of the digital word are odd, and if no errors are present in the N bits of the digital word, the digital signature comprises all ones.

33. (Previously presented) The method of claim 29 wherein if the N bits of the digital word are even, and if no errors are present in the N bits of the digital word, the second error correcting code is identical to the first error correcting code.

34. (Presently amended) The method of claim 29 wherein if the N bits of the digital word are odd, and if no errors are present in the N bits of the digital word, the second error correcting code is ~~opposite~~ the complement of the first error correcting code.

35. (Presently amended) A method for operating a memory system comprising:

A. receiving a digital word having N bits of data and M bits for error detection;

B. generating a primary error correction code based on the N bits of data of the digital word;

C. generating a complementary error correction code based on the N bits of data of the digital word;

D. performing a first logic operation on the primary error correction code and the complementary error correction code to generate a data signature representative of a comparison of the primary error correction code and the complementary error correction code;

E. determining whether an error has occurred in ~~at least one of~~ the N bits of the digital word, the primary error correction code ~~and~~ or the complementary error correction code based on the value of the data signature.

36. (Presently amended) The method of claim 35 wherein Step E includes:

performing a second logic operation on the data signature and the M bits of the digital word to generate a constant signal representing a comparison of the data signature and the M bits of the digital word;

comparing the generated constant signal to a predetermined constant signal; and

determining that an error has occurred in ~~at least one of~~ the N bits of data in the digital word, the M bits of data in the digital word, the primary error correction code ~~and~~ or the complementary error correction code if the generated constant signal is different from the predetermined constant signal.

37. (Previously presented) The method of claim 36 wherein the primary error correction code includes Y bits, wherein each one of the Y bits is generated by performing a third logic operation on predetermined bit sets of the digital word.

38. (Previously presented) The method of claim 37 wherein at least two of the Y bits of the primary error correction code are generated by performing the third logic operation on different predetermined bit sets of the digital word.

39. (Previously presented) The method of claim 38 wherein the complementary error correction code includes Y bits, wherein each one of the Y bits of the complementary error correction code is generated by performing a fourth logic operation on bits of the digital word which are not included in the predetermined bit set of the digital word used to generate a corresponding bit of the primary error correction code.

40. (Previously presented) The method of claim 37 wherein the M bits of the digital word for error correction are parity bits.

41. (Presently amended) The method of claim 37 wherein the predetermined constant signal comprises Z bits, all of which being ~~one of~~ either all zeros ~~and~~ or all ones.

42. (Presently amended) The method of claim 41 wherein the generated constant signal comprises Z bits and, if no errors are present in ~~at least one of~~ the N bits of data in the digital word, the M bits of data in the digital word, the primary error correction code ~~and~~ or the complementary error correction code, all of the Z bits of the generated constant signal are ~~one of~~ either all zeros ~~and~~ or all ones.

43. (Previously presented) The method of claim 42 wherein the first logic operation is an XOR operation.

44. (Previously presented) The method of claim 43 wherein the second logic operation is an XOR operation.

45. (Previously presented) The method of claim 44 wherein the third logic operation is one of an XOR operation and an XNOR operation.

46. (Previously presented) The method of claim 45 wherein the fourth logic operation is one of an XOR operation and an XNOR operation.

47. (Presently amended) The method of claim 39 wherein if no errors are present in the N bits of the digital word, the complementary error correcting code is ~~one of~~ either identical to the primary error correcting code ~~and opposite~~ or complement to the first error correcting code.

48. (Presently amended) The method of claim 39 wherein if no errors are present in the N bits of the digital word, the digital signature comprises ~~one of~~ either all zeros ~~and~~ or all ones.

49. (Previously presented) The method of claim 48 wherein if the N bits of the digital word are even, and if no errors are present in the N bits of the digital word, the digital signature comprises all zeros.

50. (Previously presented) The method of claim 49 wherein if the N bits of the digital word are odd, and if no errors are present in the N bits of the digital word, the digital signature comprises all ones.

51. (Previously presented) The method of claim 47 wherein if the N bits of the digital word are even, and if no errors are present in the N bits of the digital word, the complementary error correcting code is identical to the first error correcting code.

52. (Presently amended) The method of claim 47 wherein if the N bits of the digital word are odd, and if no errors are present in the N bits of the digital word, the complementary error correcting code is ~~opposite~~ the complement of the first error correcting code.

53. (Presently amended) The method of claim 35 further comprising:

F. performing a second logic operation on the data signature and the M bits of the digital word to generate a constant signal representing a comparison of the data signature and the M bits of the digital word; and

G. determining that an error has occurred in ~~at least one of~~ the N bits of data in the digital word, the M bits of data in the digital word, the primary error correction code ~~and~~ or the complementary error correction code if the generated constant signal comprises zeroes and ones.

54. (Presently amended) The method of claim 35 wherein step E comprises determining that an error has occurred in ~~at least one of~~ the N bits of data in the digital word, the primary error correction code ~~and~~ or the second error correction code if the data signature comprises zeroes and ones.

55. (Canceled)

56. (Canceled)